Title
Issues in multi-media information networks

Author
Suda, Tatsuya

Publication Date
1986

Peer reviewed

Issues in Multi-Media Information Networks

Technical Report 86-08

Tatsuya Suda
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717

Part of this research is supported by the Academic Senate Committee on Research, University of California, Irvine.

Contents

ABSTRACT

In an integrated service environment, where users exchange various types of aural and visual information, networks should appear friendly to its users providing tools for management of multi-media information. Networks should also efficiently satisfy diverse performance requirements of different information being exchanged.

In this paper we present new architecture for integrated service networks being investigated and developed by the Distributed Computation and Communication Group at the Department of Computer Science in the Columbia University. Research efforts are devoted to developing both (1) document management software to allow users to manipulate and relate text/graphics/voice information in a dynamic way, and (2) a tree network architecture for reliable and efficient exchange of multi-media information.

1. INTRODUCTION

Recent interests in the human interface to computer systems has spurred the development of integrated service workstations which facilitate manipulation of various kinds of information such as data, voice, text, graphics and facsimile, and the communication systems which will carry these various kinds of information [1 - 4]. Thus the future communication systems must be able to integrate information having various requirements and constraints and support bandwidth ranging from slow data rates to a rate of a few Mbits/sec of teleconferencing application or computer graphic application [5 - 7]. Communication systems will have to offer not only secure delivery, which is required typically in data communications, but also real time delivery for certain types of information such as voice [8]. Synchronized delivery of multiple information streams will also be required. For instance, in teleconferencing application, pictures or graphs might be required to be delivered synchronously with associated aural explanations.

Although tremendous amount of research effort has been devoted to integrated service communication systems [3], we have not found satisfactory solutions. Unfortunately, neither telephone networks nor data communication networks provide users with integrated service environment. Difficulties and problems associated with design of integrated service networks may fall into one of the following categories.

(1) Problems at the Man-Machine Interface Level, which are concerned with providing users with friendly network interface for multi-media information handling;

(2) Problems at the Network Architecture Level, which discuss network architectures which efficiently integrate and transmit various kinds of information, while satisfying requirements and constraints of each information medium.

The research project on multi-media communication systems being conducted by the Distributed Computation and Communication Group at the Department of Computer Science in the Columbia University aims to answer problems at these levels through analysis and experiments [1]. Project goals are (1) to develop a friendly user interface to a multi-media network, and (2) to design a network for efficient exchange of multi-media information.

Software to manage multi-media documents in a dynamic way is being developed on a UNIX based workstation to provide users with powerful communication tools [1]. The management system allows users to interactively manipulate and synchronize (or relate) information in diverse media: text, computer-generated graphics and aural information. Primitive version of text/graphics/voice integrated editor, which will be a part of greater multi-media information management system, has been implemented.

New network architecture, a tree network with collision avoidance switches, has been proposed and its performance has been obtained [9, 10]. A switch in this network has selection and broadcast capability to avoid collisions among simultaneously transmitted information. It has shown that the network is highly efficient in performance.

In the following sections we will discuss our research environment and details of research topics, starting with the software for management of multi-media information, followed by a tree network architecture.

2. THE MANAGEMENT OF MULTI-MEDIA INFORMATION

2.1 Research Environment

Our research environment for software development consists of UNIX based SUN workstations (Sun Microsystems Inc.) connected on an Ethernet. Those workstations are Motorola 6810 processor based machines with bit-map display, mouse (pointing device), disk (169 MBytes in our configuration), window systems and ACM-standard graphic package.

New hardware and device drivers are added to a workstation for the aid of the research in multi-media environment [11]. Voice board (based on a TI TMS 320 signal processing chip) with LPC voice coding algorithm has been added for both on-line and off-line communication purpose. Digitizer was also connected via the IEEE 796 standard parallel interface board for graphical aids to allow users to write free-hand figures. Two types of drivers have been implemented for the new hardware. One uses start/stop transmission with x-on/x-off flow control and is for slow input device. The other is for real time application and is run concurrently with user application programs. For the latter driver, UNIX kernel was also modified to meet real time requirements. These drivers are invoked by interruption (user request) and, once invoked, transfer blocks of information between I/O device and workstation memory (DMA).

2.2 Multi-Media Document Management Software

In multi-media environment, users need to manipulate information in diverse media. A multi-media document needs not be simply the storing of an existing book, article or newspaper on an electronic medium. With the aid of the computer and its ability to actively transform, construct and present information, we can communicate with others in a more dynamic setting and go one step beyond the linear, static nature imposed by the paper medium. Examples are dynamic document where users can look up references, comments, summaries, and other sources instantaneously, and interactive teaching process (computer aided instruction) coupled with the presentation of animations and pictures for clearer illustration.

The ideal system for managing multi-media documents should be able to manipulate, synchronize, and even transform information among different media, in an interactive manner. It should also be capable of manipulating several media simultaneously, and must have a friendly user interface.

As the first step towards the ideal management system, we have implemented a preliminary version of a multi-media (text/graphics/voice) editor with emphasis on 1) the cross-referencing of documents, the spatial relation and synchronization between different media of information, 2) the user interface, and 3) the structure, storage and management of data. Here, cross-referencing pertains to the capability to associate with other documents, or parts (e.g., sections) of documents. Spatial coordination and synchronization can be defined as relating objects and events (e.g., user-specified parts of documents) in space and time (Fig.1). For example, a reader of a multi-media document will see, while he is reading a particular part of the text, a graphical object associated with that part of the text (cross-reference) at the location specified by the creator of the document (spatial coordination). When the text part disappears from the display, associated graphic object also disappears (synchronization). Aural explanation which must go with graphic object or with animation should be timed correctly (synchronization).

Because this first version is intended as a test bed, we chose to include simple, easy-to-implement and semi-workable ideas. A number of these functions could have been made more efficient and sophisticated, but the goal of the first step of the project was to get a test system working quickly. Examples of short-cuts taken are the memory management routines which use a simple demand paging with prefetch technique, fixed-size data blocks for data from all the different media, and a simple message-passing facility which allows short messages to be communicated between media.

In the following, we will give a detailed description of the inner-workings of the preliminary version of the editor [1]. The user interface for different media will be discussed first, followed by the data structures and memory management schemes, and simple integration techniques.

The user interfaces for the different media is implemented in a consistent, uniform fashion (Fig.2). The mouse, as the pointing device, is used in all media, and the user interface is programmed to utilized the window, menu, scroll bar, and interactive help facilities. Menus contains commands that the user can execute. Information in a text, graphics or voice medium is displayed through a window (or view). Each window contains only one medium of information, either text, graphics or voice. The cross-referencing feature will allow the author to reference other electronic documents, and the readers to easily follow up on these cross-references and examine the related material. (Referred document or object is automatically displayed on a screen.)

Within the text medium, editing operations are similar to those found in traditional computer text editors. The keyboard is the primary source of input for data. In an attempt to demonstrate flexibility and extensibility, two user editing interfaces are being implemented for the text medium. In the first interface, the mouse may be used to position the cursor, scroll the text, cut and paste regions of text, and perform other editing functions. The second interface follows a simple traditional editing technique where the keyboard, in addition to the inputting of data, can also be used to execute editing commands. These two implementations were chosen to accommodate the two large and different classes of users: those who are comfortable editing text with a pointing device, and those accustomed to a keyboard-oriented editor. Routines that perform string operations, such as search and replacement, has been available. A modest font library has been included.

The graphics editing facility relies heavily upon the availability of a pointing device, specifically, the mouse. Basic graphics editing commands are available through pull-down and pop-up menus. The execution of commands is done through selection of menu items (commands) and interaction with graphical objects. The user may draw graphic primitives, such as line, arc, rectangle, ellipse, polygon and free-form, and perform simple cut and paste, move, copy, and erase functions. Segments may be grouped, saved as a single, named object, and stored for future reference.

For the voice medium, microphone, speakers and related analog-digital hardware provides the principal form of interface. In voice editing mode, as the voice is played for a user, a horizontal line is displayed inside a voice window. The line is a featureless representation of time. Small virtual tick marks are placed on the line and used to determine the start and stop points of the edited voice. It is possible to cut and paste a section of the voice and replay the modified speech.

This electronic document, as stored in a file on disk, contains two areas of data, the control blocks which are used to point to sections of data, and the actual text, graphics or voice data (Fig.3). An entry in the control block contains a unique identifier for an object of some type and a pointer to the block of data that describes the object. The function of a control block and its pointers can be explained through an example (reference example): "text medium references a graphics object A". In order to locate A, we search through the control blocks for an entry for A. Once found, we can follow the pointer to the first block of data that describes object A. The control blocks are primarily used for speedy access of data. However, the data being referenced, as in the example above, need not be of a different medium, like a graphics object or a track of voice data, but may be another text chapter or a cross-reference document.

The control blocks are doubly-linked for easy traversal and the last block contains a null forward pointer. When the null forward pointer is reached, we know the next block we find will be data. All the data blocks are also doubly-linked and reference pointers are kept for the beginning and the end of the link list. The doubly-linked list allows for bi-directional traversal through the data blocks and easy forward and reverse searches. The begin and end reference pointers indicate where the data section reside.

Managers are defined as global facilities used by all media. Examples include the window, display, menu, icon, and font managers. Various utilities, such as a message-passing mechanism, memory management and I/O procedures, constitute a common set of tools to be used by all media. This modular approach allows the environment to be flexible and extensible. Different routines or modules can be replaced and when adding a new medium to the environment, much of the tools, user interfaces and conventions for communicating with other media are available to the programmer.

3. A TREE NETWORK WITH COLLISION AVOIDANCE SWITCHES

In an integrated service environment, where users exchange real time voice, data and multi-media documents, networks are required to satisfy performance requirements of different information media. For instance, interactive data traffic is bursty in generation and requires small network delay [12]. Bulk data, on the other hand, consists of long messages, and requires high throughput, but real-time delivery consideration is not of its primary importance. Strict error control and recovery procedures are required in either type of data transmission. Voice calls last for a few minutes, with for more than 60% of the time the channel remains idle [13, 14]. For the integrity of the conversation, voice must be delivered within some bound (typically 50 - 100 msec); but in many data applications a delay of as much as 100 or 200 msec does not present a problem. Conversation is inherently robust, and as a result, speech can be reconstructed at the destination with acceptable quality, provided that the information loss is less than some specified fraction (typically 1 %).

As a viable alternative to the existing network architectures, a tree network with collision avoidance switches have been proposed for efficient exchange of multi-media information. In a tree network, collision avoidance switches are connected by full duplex transmission lines and form a rooted tree topology with users at its leaves. A user transmits a packet whenever he has a new one. The collision avoidance switch allows packets to go through when it is idle, and blocks them when busy (selection capability); thus collisions caused by simultaneous transmissions of packets are avoided. Blocked packets are retransmitted by the sender upon the detection of blocking, while unblocked packets, after having climbed up the tree to an appropriate height, are broadcast to a subset of users (local broadcast capability). Packets are broadcast locally to maximize the number of concurrent transmissions. In addition, collisions among packets are completely avoided, hence, the network is highly efficient in performance.

A collision avoidance switch is first introduced with only selection capability by Closs and Lee [15], and later by Albanese [16], and by Lee and Boulton [17]. They discuss hardware implementation issues of the switch along with its application for a star topology network. Routing mechanism is developed by Yemini [10] for a binary tree network with collision avoidance switches. He assumes capability at a switch to read destination address of a packet and to send the packet out to the appropriate line. A switch assumed in [10] is no longer a simple selector assumed in [15 - 17]. In the tree network proposed here we assume a collision avoidance switch with both selection capability and local broadcast capability.

In the following we first discuss transmission protocol and performance of a broadcast star network, which is the simplest case of a tree network, to illustrate how a tree network operates. Transmission protocol of a general tree network is also described.

3.1 Broadcast Star Network

In a broadcast star network, all users are directly connected to a switch, resulting in a star topology. Transmission line between a user and the switch is assumed to be full duplex; uplink to send a packet from a user to the switch and downlink to send (broadcast) a packet from the switch to a user. Up- and downlinks are of the same channel capacity.

A switch (conceptually) consists of a selection and a broadcast components (Fig.4). All up- and downlinks are connected to a selection and a broadcast components, respectively. A selection component is called busy, if a packet is currently passing through the component; otherwise it is called idle. If an idle selection component receives a packet, it sets itself busy and passes the packet to a broadcast component. Arriving packets are discarded, when the selection component is busy. When the end of transmission is detected, a selection component is set to be idle. A broadcast component broadcasts the packet from a selection component to all users. (See [15 - 17] for hardware design of the switch.)

User protocol is simple. Like in random access networks, a user transmits a packet whenever he has a new one, and starts monitoring the downlink. If the transmitting user receives broadcast of his packet after a round trip propagation delay to the switch, he knows that he has acquired the switch and his transmission is successful. Otherwise, i.e., if the user receives either broadcast of someone else's packet or nothing after a round trip propagation delay, his transmission was blocked at the switch. In this case, a user retransmits his packet immediately following the detection of a blocking. Unlike in random access networks, users do not wait for random retransmission rescheduling delay. A user repeats this process until the packet is successfully transmitted.

Analytic models have been developed for a broadcast star network, and its merit over the contention type local networks (such as Ethernet) has been proved [9]. Fig.5 shows the maximum throughput S_{max} of a homogeneous star network with 20 users. R is a round trip propagation delay from a user to the switch. S_{max} is defined as the mean number of successfully transmitted packets per one packet transmission time. Poisson arrivals and exponential transmission times of packets are assumed in the analysis. m is the mean packet transmission time, namely $m = \frac{P}{C}$, where P is the average packet length (bit) and C is the channel speed (bits/sec). Dashed line shows the throughput of an Ethernet for $m = 20$. From the figure it is shown that the throughput of a broadcast star network is much higher than that of an Ethernet. This is due to the fact that there are no collisions among packets in a broadcast star network. Even if more than two users attempt to send packets at the same time, at least one of them is guaranteed to use the network successfully. However, in an Ethernet or in any other random access networks, all packets accessing the network at the same time are destroyed and are retransmitted.

3.2 Tree Network

A tree network is an extension of a broadcast star network. By assuming an additional capability (local broadcast capability) at a switch, packets are broadcast to an appropriate subset of users so that a number of concurrent transmissions are possible in a network; thus the network is highly efficient in performance.

In a tree network, collision avoidance switches are connected by full duplex transmission lines and form a rooted tree topology with users at its leaves (Fig.6). S_i^j is the i-th switch at the level j, where the level of a switch is the hop-distance (the number of hops) to the farthest user.

Switch configuration in a tree network is shown in Fig.7. Selection and broadcast components are the same as those assumed in a broadcast star network. A selection component lets the first packet from the lower switches go through it and blocks other packets. The packet is then passed to both an upper level switch and a filter component within the switch (through a broadcast component). The filter component checks the header of the packet to see if the level indicated in the header matches its own level (i.e, the level of the switch). (As we will see later, packet header includes the level of the switch at the root of the minimum subtree containing both origin and destination users of the packet.) If not, the filter component simply discards the packet. If it does, the packet is passed to a downlink selection component.

A downlink selection component acts in exactly the same way as a selection component in a broadcast star network except it always puts priority on packets from its upper level switch over those from the filter component. Namely, whenever the downlink selection component receives a packet from the upper switch, it aborts the current transmission, if there is, and lets the packet from the upper switch go through. Transmission of a packet from the filter component will be completed if and only if there is no packet coming down from the upper switch during its transmission. Packets are then broadcast to lower switches through a broadcast component.

User protocol is simple. When user has a packet to send, he attaches a header containing the level of the switch at the root of the minimum subtree which contains both himself (i.e., origin) and the destination user. (Network topology is assumed to be known to all the users.) User transmits the packet and starts monitoring the downlink for the duration of R, where R is the round trip propagation delay to the tree top switch (, not to the root switch of the subtree). If the user has received a complete broadcast of his packet within R (in case that the packet is shorter than R), or if he is still receiving a broadcast of his packet after R (in case that the packet is longer than R), he knows he has acquired the network channel capacity and his packet transmission is successful. Otherwise, (i.e., if he receives either another user's packet or his own packet truncated in the middle, or receives broadcast of no packets, within R), the user knows his transmission was unsuccessful. In this case, the user retransmits the packet immediately. A user repeats this process until the packet is successfully transmitted.

The following worst-case scenario explains why a user must wait for a round trip propagation delay to be guaranteed his use of network capacity. Suppose that user 4 wants to communicate with user 6 in Fig.6, and further, the round trip propagation delay of user 4 to the tree top switch S_1^3 is R. At time t_0 user 4 begins transmitting his packet (test packet). At time just before the test packet arrives at the tree top switch, namely, at time $t_0 + R/2 - \epsilon$, where ϵ is an infinitesimally small value, suppose a packet destined to user 5 from user 1 arrives at an idle tree top switch. This packet is broadcast to all users and eventually interrupts transmission of the test packet at the switch S_1^2, S_2^1 and S_3^1. Broadcast of the packet from user 1 will reach user 4 at time $t_0 + R - \epsilon$. Thus, new arrivals at other users in the network during the round trip propagation delay may cause a blocking of the current transmission, and a user cannot be sure whether it has seized the channel until the round trip propagation delay passes since the beginning of packet transmission. However once the channel has been seized, no other users can interfere with the transmission.

The above example also shows why a packet must be passed up to the tree top switch, even if the root switch of the appropriate subtree is lower than the tree top switch. The packet is transmitted to the tree top to make the switches on its way busy so that the subtree is logically disconnected from the rest of the network. Hence, packets from the outside of the subtree cannot interfere with the transmission within the subtree. In other words, transmission in one subtree does not interfere transmissions in other subtrees, and thus a number of concurrent transmissions are possible.

As discussed in a broadcast star network, even if more than two users attempt to send packets at the same time, at least one of them is guaranteed to use the network successfully due to selection capability of switches. This easily leads to the observation that performance of a tree network is better than any random access networks, where all packets accessing the network at the same time are destroyed (i.e., collided) and are retransmitted. Furthermore, packets are broadcast locally, and thus multiple concurrent transmissions are possible. Communications within a subset of users might scarcely disturb those within another subset of users.

4. CONCLUSIONS

In this paper we discussed issues in multi-media communication systems and described research project being developed at Columbia University. Our research focusses on 1) design and analysis of network architecture and 2) multi-media information management system.

REFERENCES

[1] T. Suda, Y. Yemini, B. Kwartler, L. Hu and A. Yoshimoto, "Multi-Media Communication Systems", Proc. of the IEEE International Conference on Communications (ICC), pp.13.5.1-13.5.5, 1985.

[2] Special Issue on Engineering Workstations, the IEEE Computer Graphics and Applications Magazine, Vol.4, No.8, Aug. 1984

[3] Special Issue on Integrated Service Digital Networks, the IEEE Communications Magazine, Vol.22, No.1, Jan. 1984.

[4] A. G. Solosky, "iNet 2000 : An Intelligent Communication Service", Proc. of the ICC, pp.93-97, 1984.

[5] L. A. Parker and C. H. Olgren, Ed., "The Teleconferencing Resource Book", North-Holland, 1984.

[6] E. F. Brown, G. L. Cash and O. Johnsen, "An Interactive Graphics Conferencing Test-Bed", Proc. of the ICC, pp.728-733, 1984.

[7] C. F. C. Greinacher, E. Bach, K. Muller and K. Patzelt, "A Realistic Approach to Evaluating Digital Imaging Systems", Proc. of SPIE Application of Optical Instrumentation in Medicine XII, pp.91-98, 1984.

[8] B. Gold, "Digital Speech Network", Proc. of the IEEE, Vol.65, No.12, pp.1636-1658, Dec. 1977.

[9] T. Suda, Y. Yemini and M. Schwartz, "A Tree Network with Collision Avoidance Switches", Proc. of the INFOCOM, pp.105-113, 1984.

[10] Y. Yemini, "Tinkernet : Or, Is There Life Between Lans and Pbxs?", Proc. of the ICC, pp.F3.1.1-F3.1.5, 1983.

[11] T. Suda and Y. Yemini, "Integrated Service Distributed Computer Systems", Technical

Report, Department of Computer Science, Columbia University, June, 1983.

[12] L. Kleinrock and W. E. Naylor, "On Measured Behavior of the ARPA Network", AFIPS Conf. Proc. of the National Computer Conference (NCC), pp.767-780, 1974.

[13] D. Cohen, "Packet Communication of Online Speech", Proc. of the NCC, pp.169-176, 1981.

[14] P. M. Gopal, J. W. Wong and J. C. Majithia, "An Evaluation of Playout Strategies for Voice Transmission in Packet Networks", Proc. of the Computer Networking Symposium, pp.33-38, 1981.

[15] F. Closs and R. P. Lee, "A Multi-Star Broadcast Network for Local-Area Communication", IFIP Proc. of Local Networks for Computer Communications, pp.61-80, North Holland, 1981.

[16] A. Albanese, "Star Network with Collision-Avoidance Circuits", the Bell System Technical Journal, Vol.62, No.3, pp.631-638, March 1983.

[17] E. S. Lee and P. I. P. Boulton, "The Principles and Performance of Hubnet: A 50 Mbit/s Glass Fiber Local Area Network", IEEE Journal on Selected Areas in Communications, Vol.SAC-1, No.5, pp.711-720, Nov. 1983.

Figure Captions

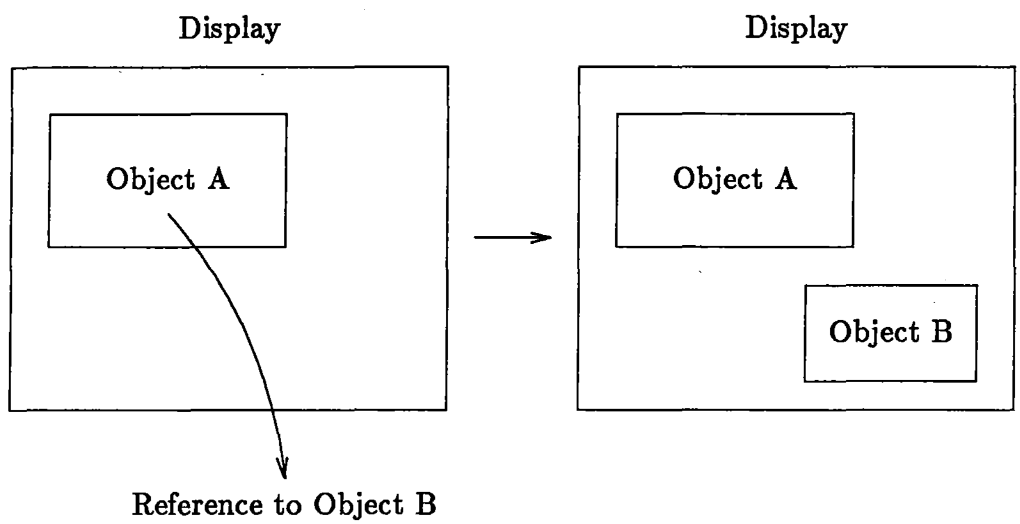

Fig.1 Spatial Coordination and Synchronization

Display

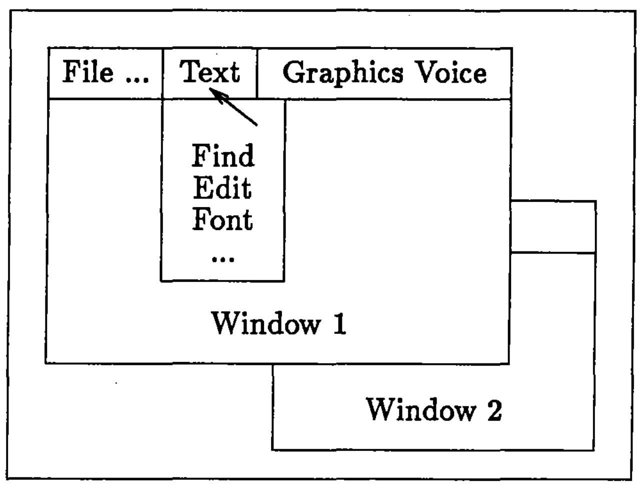

Fig.2 User Interface

File

	Main Text	Location nn	Control Block
	Object A	Location aa	
	Object B	Location bb	
	...	...	
nn	Header : main text, block size		Data Block
	This contains main body with references to various objects.		
aa	Header : object A, block size		
	object A data		
bb	. . .		

Fig.3 File Structure

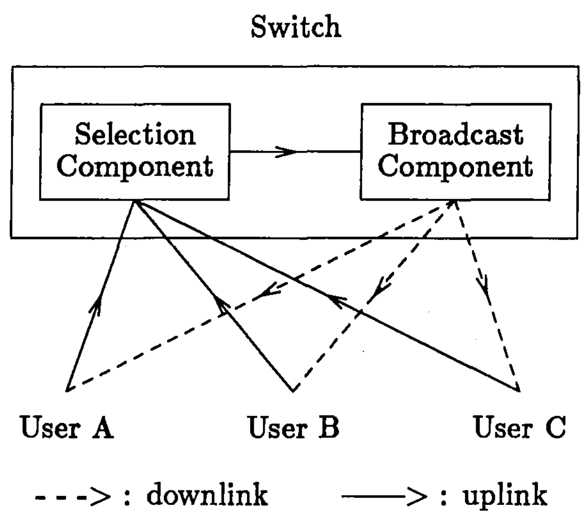

Fig.4 Switch Configuration in a Broadcast Star Network

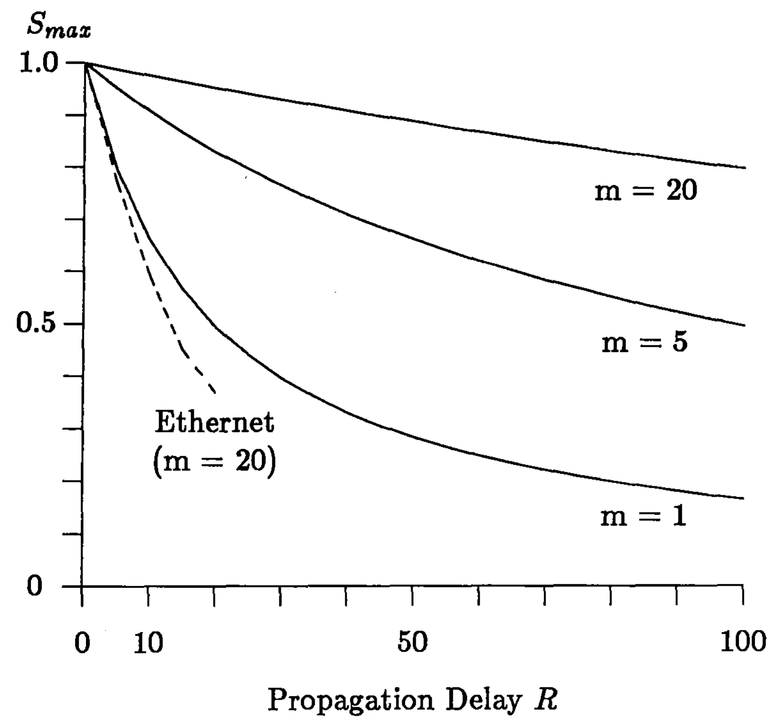

Fig.5 Throughput of a Broadcast Star Network

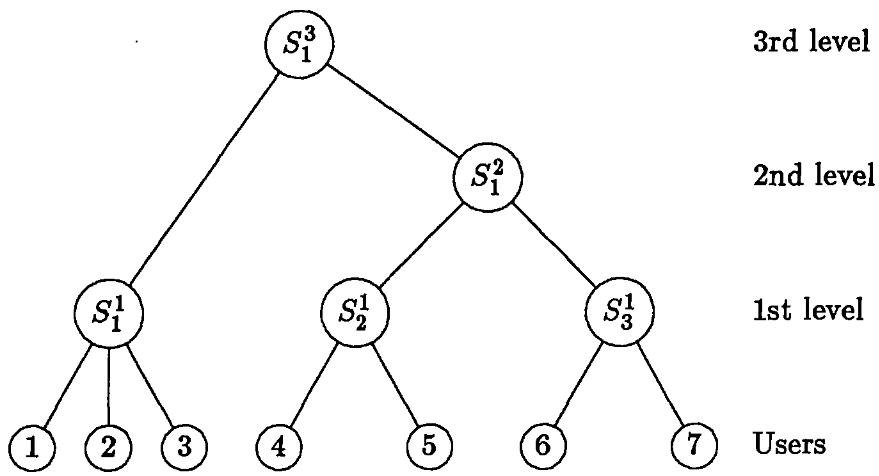

Fig.6 Topology of a Tree Network

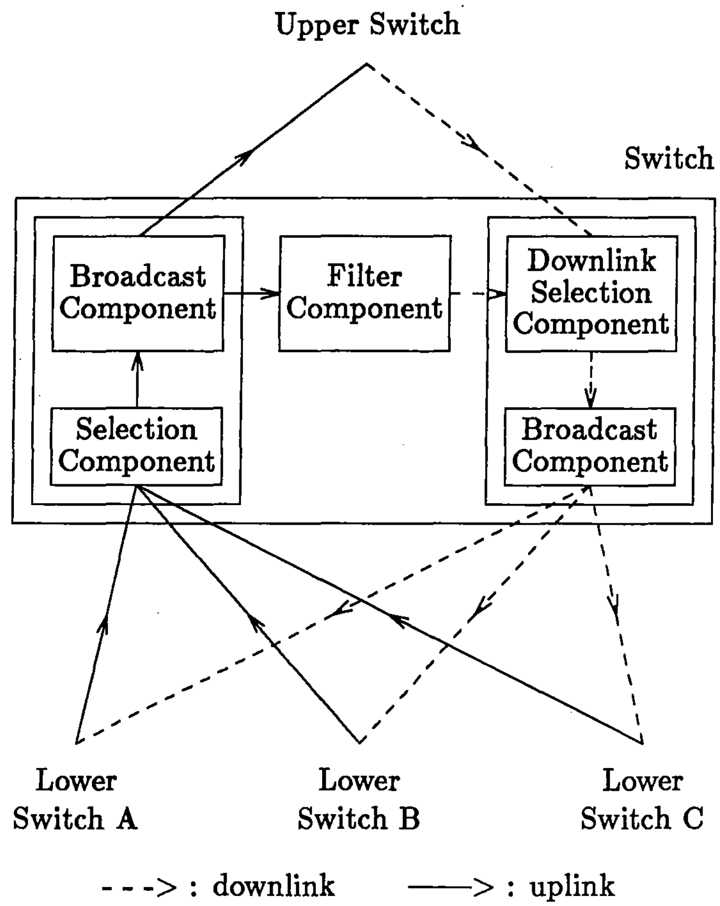

Fig.7 Switch Configuration in a Tree Network